Carnegie Mellon
Software Engineering Institute

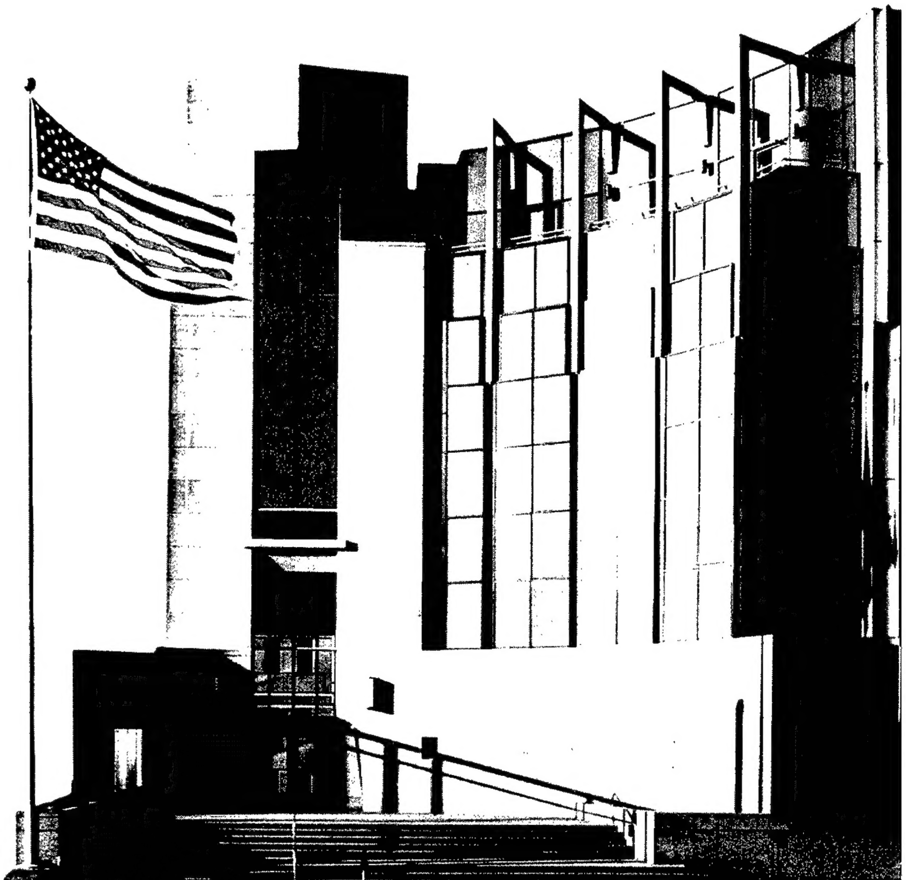
Software Product Line Adoption Roadmap

Linda M. Northrop

September 2004

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

TECHNICAL REPORT
CMU/SEI-2004-TR-022
ESC-TR-2004-022



Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Software Product Line Adoption Roadmap

CMU/SEI-2004-TR-022
ESC-TR-2004-022

Linda M. Northrop

September 2004

Product Line Practice Initiative

This report was prepared for the

SEI Joint Program Office
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Christos Scondras
Chief of Programs, XPK

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

List of Tables

Acknowledgements

All the members of the Product Line Practice Initiative at the Carnegie Mellon® Software Engineering Institute (SEI) have been contributors to the SEI body of knowledge on software product lines. The concepts described in this report draw heavily on that rich reservoir. In particular, Paul Clements and I have worked very closely on the content, organization, and orchestration of the SEI *Framework for Software Product Line Practice*^{SM} [Clements & Northrop 04] and its associated case studies and product line practices patterns [Clements & Northrop 02a]. It is impossible to tease apart our individual contributions. I am indebted to Paul for his continued partnership with me on this work. Larry Jones and I have been the primary authors of the SEI Product Line Quick Look^{SM} (PLQL^{SM}), the SEI Product Line Technical Probe^{SM} (PLTP^{SM}), and the SEI Adopting Software Product Lines course. He has provided bountiful input and helpful review on the contents of this report. Finally, I acknowledge the members of the product line adoption working groups at two successive Dagstuhl Seminars on Product Families in April 2001 and April 2003. Through discussions at Dagstuhl and subsequent collaborations on two resulting papers on product line adoption, those working group members provided another important source of information [Böckle et al. 02, Bühne et al. 03].

® Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

SM Framework for Software Product Line Practice, Product Line Quick Look, PLQL, Product Line Technical Probe, and PLTP are service marks of Carnegie Mellon University.

Abstract

The tremendous benefits of taking a software product line approach are well documented. Organizations have achieved significant reductions in cost and time to market and, at the same time, increased the quality of families of their software systems. However, to date, there are considerable barriers to organizational adoption of product line practices and to widespread product line practice. Phased adoption is attractive as a risk reduction and fiscally viable proposition. This report introduces a variant of the Factory pattern called the Adoption Factory pattern that provides a generic roadmap to guide a manageable, phased product line adoption strategy. In addition, the report examines the Adoption Factory pattern from multiple useful views and describes how it can be used. The report concludes with a summary of the Carnegie Mellon® Software Engineering Institute's experiences with the pattern to date and its future plans with regard to the pattern.

1 Introduction

Product line adoption involves moving from some form of developing software-intensive systems with a single-system mentality to developing them as a software product line. A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [Clements & Northrop 02a]. The adoption objective is to have a core asset base, supportive processes, and organizational structures; to develop products from that asset base in a way that achieves business goals; and to institute mechanisms to improve and extend the software product line adoption effort as long as it makes sense.

The tremendous benefits of taking a software product line approach are well documented [Clements & Northrop 02a]. Organizations have achieved significant reductions in cost and time to market and, at the same time, increased the quality of families of their software systems. However, for many other organizations, there are considerable barriers to adoption of product line practices [Northrop 02]. Time to devote to product line activities and the initial associated cost of doing so are the most significant barriers. Building the core asset base requires a nontrivial investment. The technical and management artifacts in the asset base must capitalize on the commonality among products and, at the same time, support the variability among those same products. Moreover, the development of the product line infrastructure—including new practices, processes, and tool support—also requires investment. Many organizations struggle with how to allocate the necessary start-up funds. And unfortunately, cost isn't the only impediment to product line adoption. Other barriers can also prove challenging and, in some cases, insurmountable. Incompatible development processes, lack of necessary knowledge and possibly talent, cultural resistance, lack of management support, incompatible acquisition practices, lack of disciplined management and development practices, lack of a product line vision, lack of a product line adoption plan, lack of a product line champion, and unrealistic expectations are among the many other roadblocks that organizations face. Even the reactive approach to software product lines described by Krueger [Krueger 02] and the incremental approach that others have proposed [Muthig 02]—though both approaches drive down the cost of adoption—are not free of adoption barriers.

An organization that cannot overcome its barriers to product line adoption will not succeed. An organization that does not know what is necessary to succeed with software product lines is unlikely to succeed, at least in a cost-effective, timely way. An organization that does not know how to go about product line adoption is unlikely to succeed, at least not without great difficulty and cost. The Carnegie Mellon[®] Software Engineering Institute (SEI) created the Product Line Practice Initiative to help such organizations. The purpose of this initiative has, from the outset, been to make product line practice a low-risk, high-return proposition for all developers and acquirers. Its major contribution to fulfill this objective has been the SEI *Framework for Software Product Line Practice*[SM] (henceforth referred to as the Framework), which describes the 3 essential activities and 29 practice areas necessary for mastery of a software product line approach [Clements & Northrop 04]. The Framework has proven to be a useful reference model used by organizations worldwide.

The "Launching and Institutionalizing" practice area of the Framework lays out what needs to occur in organizational adoption as well as some specific practices that have proven useful to multiple organizations. However, many organizations still struggle with how to apply 29 practice areas, where to start, and how to plan their product line adoption path. We learned this firsthand when we initially performed the SEI Product Line Technical Probe[SM] (PLTP[SM]) [Clements & Northrop 02a] at several organizations. The results of the PLTP include an organization's strengths and challenges related to the 29 practice areas in the Framework. Knowing baseline status with regard to the practices areas proved to be a tremendous help to organizations. However, without a higher level framing or further assistance, organizations were hard-pressed to know where to begin to attack their challenges and marshal a product line adoption effort.

Others have also recognized the difficulty of product line adoption. Böckle and associates studied software product line adoption and institutionalization needs from an organizational standpoint [Böckle et al. 02]. Bosch has examined the maturity and evaluation of product line artifacts. However, that examination does not take into account the process and business dimensions that are critical to successful product lines [Bosch 02]; product line artifact maturity does not translate to organizational product line capability. The "Crossing the Chasm" panel at the Second Software Product Line Conference (SPLC2) in 2002 discussed adoption and transition challenges—challenges that face individual product line projects, organizations, and the emerging software product line community. Bühne and associates explored the context for product line adoption at multiple levels and proposed how this characterization could be helpful in choosing an appropriate product line approach [Bühne et al. 03]. Van der Linden and associates have offered an evaluation model for software product lines of embedded systems that is centered on four axes: business, process, architecture, and organization [van der Linden et al. 04]. Many organizations have clamored for a maturity model for software product lines. It is our belief that the software product line community is too immature to confidently put forth a maturity model for product lines. Moreover, when the community is sufficiently mature, the different nature of individual organization's products, markets, and approaches may make such a model extremely difficult to implement.

[®] Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

[SM] Framework for Software Product Line Practice, Product Line Technical Probe, and PLTP are service marks of Carnegie Mellon University.

Others have written about economic models to justify product line efforts and the connection of such models to adoption [Clements et al. 04, Schmidt & Verlage 02]. None of these, however, provide a clear, generic roadmap to software product line adoption.

In principle and practice, a generic product line adoption roadmap has appeal. Organizations could follow a proven path, adapt it to their own needs, and do so in an incremental, manageable way that reduces risks. We have developed such an adoption roadmap as a variant of the Factory pattern. The Factory pattern is the most comprehensive member of the SEI collection of product line practice patterns [Clements & Northrop 02a]. Our variant is called the Adoption Factory pattern.

This report describes and analyzes the Adoption Factory pattern and its utility. Section 2, which follows this introduction, reviews the concept of product line practice patterns and, in particular, the Factory pattern. Section 3 describes the Adoption Factory pattern—our product line adoption roadmap—and examines it from multiple useful views. Various uses of the Adoption Factory pattern are presented in Section 4, and Section 5 offers a conclusion that summarizes our experiences with the pattern to date and our future plans. The appendices provide further details about the SEI product line practice patterns.

2 The Factory Pattern

Patterns are a way of expressing common contexts and problem/solution pairs. Patterns have been useful in many disciplines and popularized especially among software developers in the form of software design patterns [Gamma 95] and software architecture patterns [Bruckhauas 96], which have both become part of the mainstream software developers' vocabulary.

2.1 Product Line Practice Patterns

Following this trend, we defined a collection of product line practice patterns [Clements & Northrop 02a]. Software product line practice patterns

- address recurring product line problems that arise in specific software product line situations and present solutions to them
- document existing, well-proven software product line experience
- identify and specify abstractions that are broader in scope than single practice areas
- provide an additional common vocabulary and understanding for software product lines
- are a means of documenting new software product line efforts
- help manage the complexity inherent in software product line approaches
- can be combined to build complex product line solutions

Our collection of 12 patterns and 10 variants characterize common product line contexts and problem/solution pairs that we have observed. Each pattern is described using a common template and follows the general context-problem-solution schema illustrated in Figure 1.

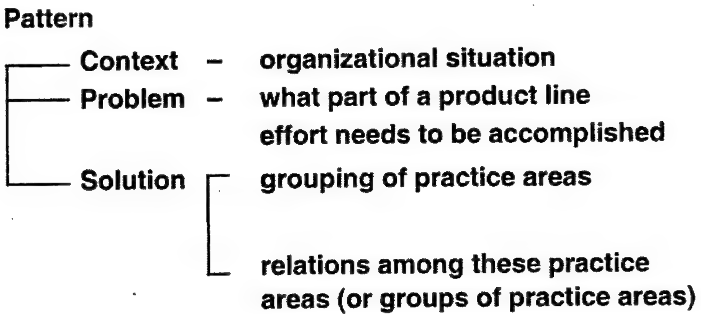

Pattern
Context – organizational situation
Problem – what part of a product line effort needs to be accomplished
Solution
grouping of practice areas
relations among these practice areas (or groups of practice areas)

Figure 1: Schema for Software Product Line Practice Patterns

Appendix A lists the patterns and variants in the SEI collection. The product line practice patterns span various ranges of abstraction, scale, and purpose. The context for some of the patterns is universal—that is, they apply in all situations. The context for other patterns is particular to specific organizational conditions. Some of the patterns are related in that they solve a part of the overall product line approach and that a pattern hierarchy makes sense. Our direct experience and the feedback we have received attest to the usefulness of the patterns in packaging solutions to particular product line problems. We don't pretend that our collection is complete. A pattern collection is intended to be grown by a community. We are hopeful that the product line community will add to and improve our collection as the experience with software product lines grows.

2.2 A Pattern for the Entire Product Line Organization

The Factory pattern is one of the product line practice patterns in our collection. It is a composite pattern that describes the entire product line organization. It provides a picture of what an organization would look like if it had product line capability. The Factory pattern describes fielding a product line as accomplishing the following six tasks:

1. deciding what products you wish to build
2. building and running the production capability (the assembly line, if you like) to build those products
3. preparing the organization to effectively use the assembly line
4. designing and providing the parts that will roll down the assembly line to be joined together to form the product
5. running the assembly line and building products from those parts
6. monitoring the process, keeping a pulse on the operations, and applying corrections as necessary to keep the organization on course

Accordingly, the Factory pattern consists of the subpatterns described in Table 1.

Table 1: Subpatterns in the Factory Pattern

Subpattern	Description
What to Build	yields the set of products to be included in the product line along with an associated business case
Each Asset	provides individual core assets and their attached processes
Product Parts	supplies the core assets from which products will be built
Assembly Line	provides the production infrastructure
Product Builder	yields the individual products in the product line
Cold Start	prepares the organization for its first product line operation
In Motion	keeps the product line organization running
Monitor	keeps watch on the organization and responds with any needed changes

Appendix B contains a diagram of the dynamic structure of each pattern listed above. Figure 2 illustrates the dynamic structure of the Factory pattern.

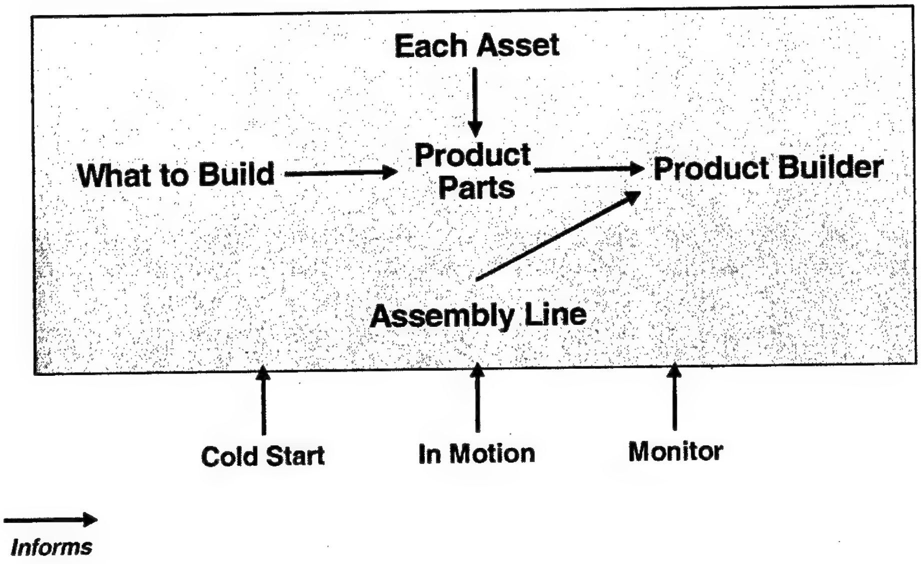


Figure 2: Dynamic Structure of the Factory Pattern

The Factory pattern offers an abstraction of the entire product line organization—a high-level view and a blueprint for a "divide and conquer" strategy. Without the Factory pattern, such a blueprint is not immediately intuitive to organizations on the threshold of a product line effort or, for that matter, those in the throes of trying to achieve a software product line approach. We have found that organizations find the Factory pattern very useful in describing their product line activities and to assigning roles and responsibilities to those involved. It seemed natural to us that the Factory pattern could serve as the basis for the clear product line adoption roadmap we were seeking. As described in the next section, that intuition proved to be true.

3 The Adoption Factory Pattern

A generic product line adoption roadmap should contain the fundamental milestones in any product line adoption effort and their dependencies. The Factory pattern meets these criteria. It comprises subpatterns and therefore provides an inherent abstraction and chunking of the major product line milestones or solution packages. Its dynamic structure (shown in Figure 2) provides a dependency ordering as depicted by the arrows denoting an "informs" relation[1] between the subpatterns. Moreover, it applies to any organization.

Our experience dictated, however, that we needed to make one small modification. Organizations that lack the ability to define and follow processes, even lightweight or agile ones, need to address that deficiency early in their adoption path. So, even though the "Process Definition" practice area (which is "about an organization's capability to define and document processes" [Clements & Northrop 2b]) is part of both the Each Asset and Assembly Line patterns, we felt it necessary to add the "Process Definition" practice area as a separate element to the Factory pattern. In this way, we arrived at the definition of the Adoption Factory pattern, a variant of the Factory pattern, shown in Figure 3.

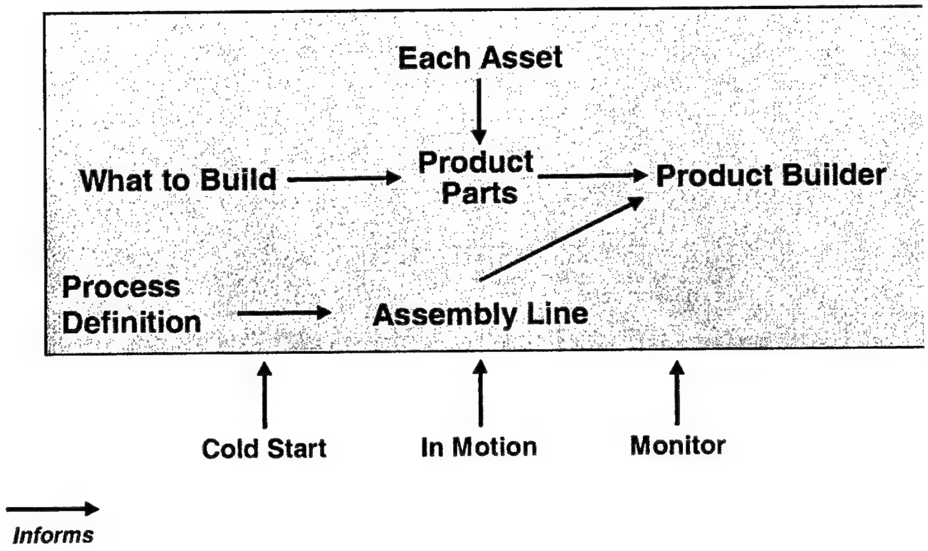


Figure 3: Dynamic Structure of the Adoption Factory Pattern

[1] The "informs" relation depicted by the arrow in this technical report's dynamic structure diagrams does not imply a strictly linear completion sequence but rather a shift in emphasis. Iteration is inherent among practice areas and subpatterns.

3.1 Description

The standard template for the Adoption Factory pattern is given below. Because Adoption Factory is a variant of the Factory pattern, the template descriptions of both patterns are quite similar [Clements & Northrop 02a, p. 393].

Name: The Adoption Factory pattern is a composite pattern that describes the milestones in any product line adoption effort and their dependencies.

Example:

Scenario 1: The chief technical officer (CTO) of a company that builds medical scanners is overwhelmed by developing and managing the software configurations for the multitude of types and versions of scanners his company produces. He recognizes that even though the software for each type of scanner has some unique features, all the software shares a significant number of common features and similar underlying fundamental tasks and behavior needs. He studied the software product line approach adopted by some of his company's competitors and knows that he needs to implement such an approach in order to wrest control over his many software products and stay competitive. However, he can't afford to proceed down the wrong path. He needs a roadmap that shows him what to do and when to do it.

Scenario 2: The software development manager of a robot manufacturer has launched an initial product line effort for the software in its line of warehouse robots. He started by defining a software architecture for the entire family of robots. The architects are struggling with the amount of variability they have to contend with, and the developers are not used to following the dictates of an architecture, much less a common one. He is wondering if there would have been a better way to begin product line adoption and would like some guidance as to how organizations should proceed, what activities he might have missed, and what midcourse corrections he can take.

Context: An organization is fielding a product line for the first time.

Problem: To provide a roadmap for its product line adoption effort

Solution: A product line adoption roadmap must include the following seven major activities:

1. deciding and justifying what products to include in the product line
2. defining, documenting, and following processes for software development and management
3. preparing the organization for a software product line approach

4. designing and providing the common assets that will be used to construct the products in the product line
5. building and using the production infrastructure (necessary plans, processes, and tools)
6. building products from the core assets in a prescribed way
7. monitoring the product line effort, keeping a pulse on the adoption activities and the product line operations, and applying course corrections as necessary to keep the organization on course

Static: The Adoption Factory pattern consists of the "Process Definition" practice area and the following subpatterns:

- Assembly Line
- Cold Start
- Each Asset
- In Motion
- Monitor
- Product Builder
- Product Parts
- What to Build

Dynamics: Figure 3 illustrates relations among the elements in the Adoption Factory pattern and defines an inherent dependency ordering.

- What to Build yields the set of products to be included in the product line along with an associated business case.
- Process Definition provides capability to define and document processes.
- Cold Start prepares the organization for its first product line operation.
- Each Asset provides individual core assets and their attached processes.
- Product Parts supplies the core assets from which products will be built.
- Assembly Line provides the production infrastructure.
- Product Builder yields the individual products in the product line.
- In Motion keeps the product line organization running.
- Monitor keeps watch on the organization and responds with any needed changes.

Application: The Adoption Factory pattern can be used as a generic product line adoption roadmap for any organization attempting a product line approach for the first time. It can serve as the basis for an adoption model for a phased software product line or product line adoption plan. The CTO in scenario 1 can use this pattern as a basic template for planning product line adoption activities. He can expand the individual subpattern definitions with their included practice areas and build dependent action plans to tackle individual practice areas or groups of them in their order of critical dependency. The software development manager in scenario 2 can use the Adoption Factory pattern to locate the "Architecture Definition" and "Component Development" practice areas in the Product Parts pattern and recognize that his organization might be suffering from an undefined or ill-defined product line scope, lack of defined processes and the ability to follow them, or lack of management and organizational support. These problems would have been addressed by the What to Build, Process Definition, and Cold Start patterns, respectively—all necessary informants to the Product Parts pattern. He can study what should have been done and then determine a remedial course of action that covers the elements of the Adoption Factory pattern that his organization skipped.

Variants: One can easily imagine variations of the Adoption Factory pattern based on known and newly defined variants of the patterns it contains.

Consequences: The Adoption Factory pattern can be used as a generic product line adoption roadmap. It provides the necessary abstraction of the major activities involved and their dependencies. In addition, by decomposing its subpatterns into their composite practice areas, a more detailed adoption plan and dependent action plans can readily be developed. Even though there are "informs" relations that move from left to right in the dynamic structure of the Adoption Factory pattern, note that the relations in the product line practice patterns are never strictly linear. Owing to the highly iterative nature of product line adoption and operations, the arrows should always be interpreted as denoting a shift in active emphasis but by no means exclusion.

It is important to underscore the content of the Consequences section of the template. No product line effort follows a waterfall life cycle, so it would be misleading to suggest that the practices in the What to Build pattern be completed only once and the results carved in stone. More realistically, the What to Build pattern activities will be completed, and emphasis will shift to growing or mining the core asset base as described by the Product Parts pattern. However, because the scope and business case may need to be refined as the core asset activities proceed, practices from the What to Build pattern may be later repeated but most likely on a much smaller scale.

3.2 Useful Views

Though the Adoption Factory pattern provides a helpful high-level roadmap, when using it to plan, analyze, and implement an organization's specific product line adoption activities, it is convenient to portray the roadmap from multiple different perspectives or views. We have found the following six views, which are described in the next six subsections of this report, to be especially helpful: (1) Adoption

Phases, (2) Focus Areas, (3) Phases and Focus Areas, (4) Practice Areas, (5) Outputs, and (6) Roles.

3.2.1 Adoption Phases

We can view the dynamic structure of the Adoption Factory pattern as three columns that designate the temporal phases of product line adoption as follows:

1. **Establish Context:** involves paving the way for the product line adoption by determining the scope and associated business case, ensuring the necessary process capability, and performing the necessary organizational management tasks. The What to Build and Cold Start patterns and the "Process Definition" practice area are in this phase.
2. **Establish Production Capability:** involves developing the core asset base and the production infrastructure, and effectively managing those efforts at project and cross-project levels. The Each Asset, Product Parts, Assembly Line, and In Motion patterns are in this phase.
3. **Operate Product Line:** involves using the core asset base to efficiently build products and effectively monitoring and improving the product line operation. The Product Builder and Monitor patterns are in this phase.

Figure 4 illustrates the phases of the Adoption Factory pattern.

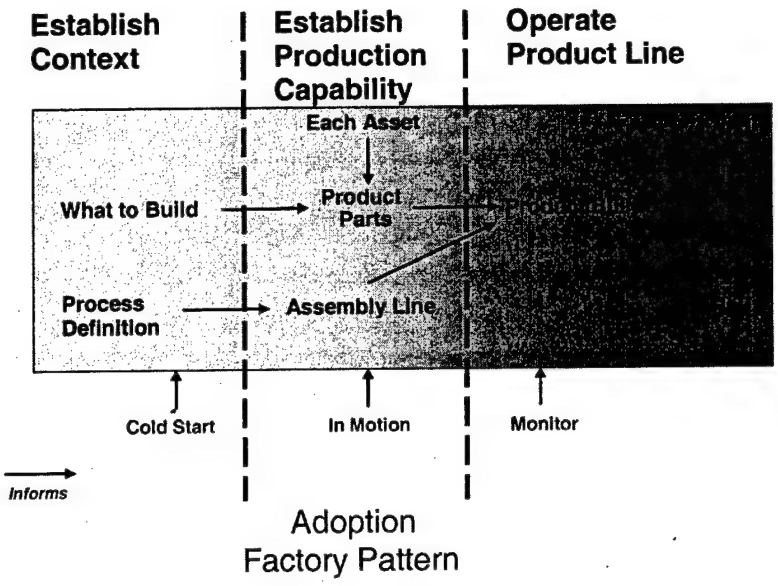


Figure 4: Adoption Phases

It is important to reiterate that even though the phases represent a temporal ordering of adoption activities, there will be iteration. Though the emphasis of effort will shift as an organization moves from an earlier to a later phase, many of the earlier phase's activities will be repeated later—usually on a much smaller scale.

3.2.2 Focus Areas View

We can also view the dynamic structure of the Adoption Factory pattern as three rows, each one corresponding to a focus area for certain patterns and practice areas. The focus areas are

1. **product:** involves those activities for defining and developing products and their common assets. The What to Build, Each Asset, Product Parts, and Product Builder patterns make up the product focus area.
2. **process:** involves the underlying processes and production infrastructure necessary to adopt a product line approach. The "Process Definition" practice area and the Assembly Line pattern constitute the process focus area.
3. **organization:** involves the management practices and activities necessary to adopt a product line approach and operate a software product line. The Cold Start, In Motion, and Monitor patterns make up the organization focus area.

Figure 5 illustrates the focus areas of the Adoption Factory pattern.

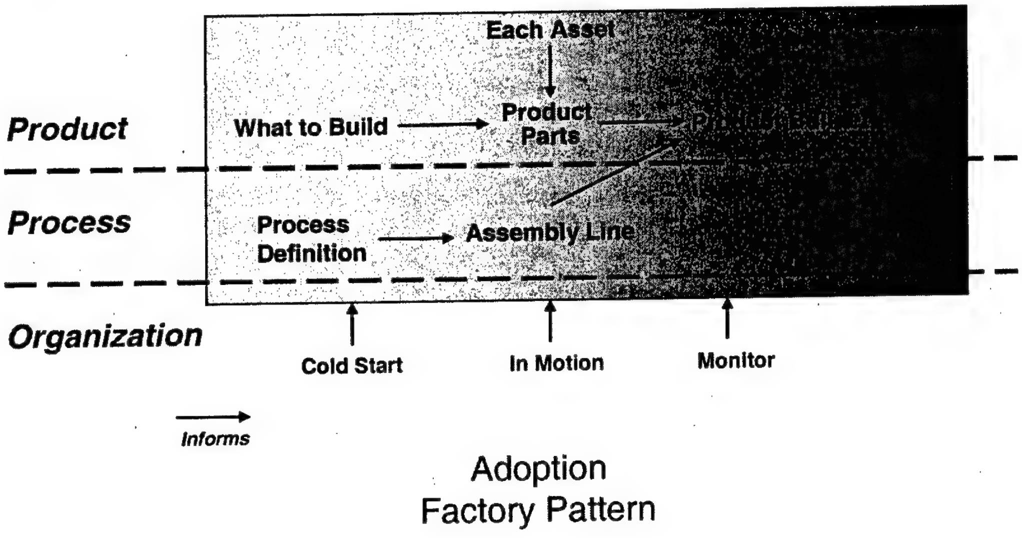


Figure 5: Focus Areas

3.2.3 Phases and Focus Areas View

We have found it especially useful to view the Adoption Factory pattern as phases and focus areas simultaneously. Figure 6 provides this perspective with horizontal focus area delineations and vertical phase delineations.

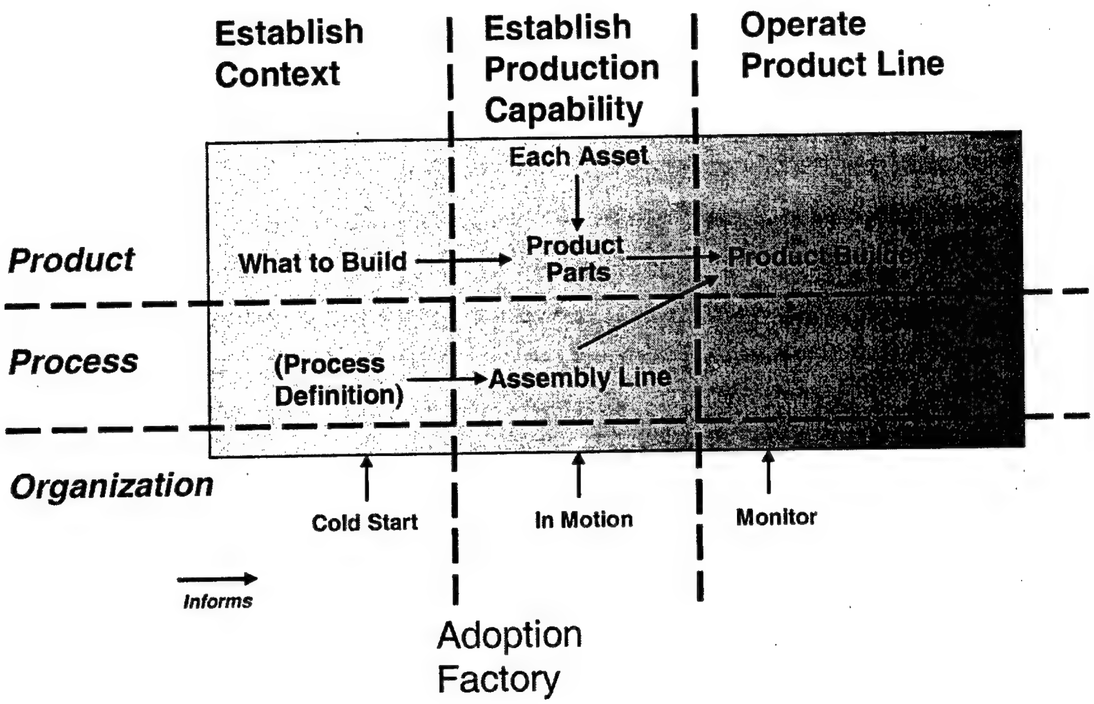
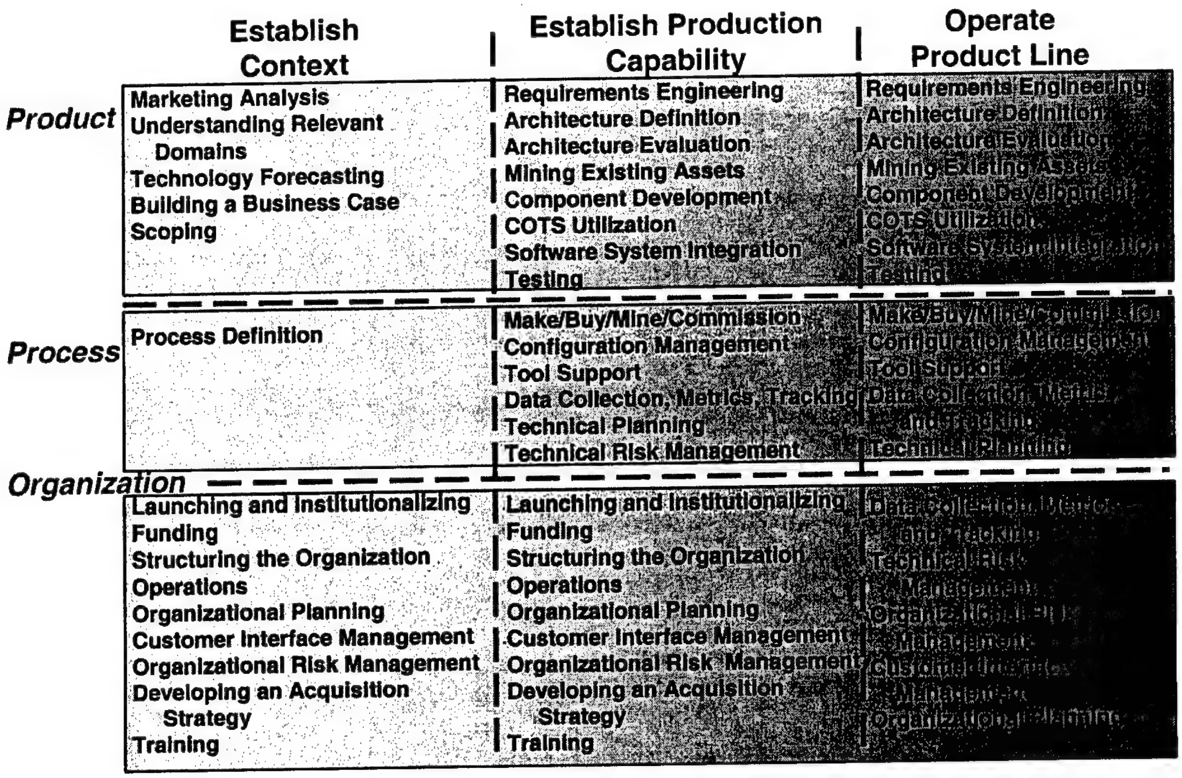


Figure 6: Adoption Factory Pattern Annotated with Adoption Phases and Focus Areas

3.2.4 Practice Areas View

The detail beneath the subpatterns of the Adoption Factory pattern (as articulated by the practice areas associated with each one) is necessary for detailed product line adoption planning. Figure 7 shows the Adoption Factory pattern and its constituent practice areas elaborated in a view that also shows the focus areas and adoption phases.



	Establish Context	Establish Production Capability	Operate Product Line
Product	Marketing Analysis Understanding Relevant Domains Technology Forecasting Building a Business Case Scoping	Requirements Engineering Architecture Definition Architecture Evaluation Mining Existing Assets Component Development COTS Utilization Software System Integration Testing	Requirements Engineering Architecture Definition Architecture Evaluation Mining Existing Assets Component Development COTS Utilization Software System Integration Testing
Process	Process Definition	Make/Buy/Mine/Commission Configuration Management Tool Support Data Collection, Metrics, Tracking Technical Planning Technical Risk Management	Make/Buy/Mine/Commission Configuration Management Tool Support Data Collection, Metrics and Tracking Technical Planning
Organization	Launching and Institutionalizing Funding Structuring the Organization Operations Organizational Planning Customer Interface Management Organizational Risk Management Developing an Acquisition Strategy Training	Launching and Institutionalizing Funding Structuring the Organization Operations Organizational Planning Customer Interface Management Organizational Risk Management Developing an Acquisition Strategy Training	Data Collection, Metrics and Tracking Technical Risk Management Organizational Risk Management Customer Interface Management Organizational Planning

Figure 7: Adoption Factory Pattern and Its Associated Practice Areas

Notice that some practice areas appear in multiple phases. However, the actual practices will vary dependent on the phase and overall objective of the associated pattern. For example, the "Architecture Definition" practice area in the Establish Production Capability Phase is associated with the Product Parts pattern and therefore involves defining the product line architecture that will be the structure of all products. The "Application to Core Asset Development" section of the "Architecture Definition" practice area's description in the Framework contains relevant guidance. However, the "Architecture Definition" practice area in the Operate Product Line Phase is associated with the Product Builder pattern and therefore involves instantiation of product architecture from the product line architecture. In this case, the "Application to Product Development" section of the "Architecture Definition" practice area's description in the Framework contains relevant guidance.

3.2.5 Outputs View

Another useful and more detailed perspective of the Phases and Focus Areas view can be obtained by listing the outputs generated in each of the nine quadrants. Table 2 lists some typical outputs that can be produced by phase and focus area.

Table 2: Outputs in the Adoption Phases

	Establish Context Phase	Establish Production Capability Phase	Operate Product Line Phase
Product outputs	• marketing description • domain model • technology survey • economic model • business use cases • cost/benefit model • business case • scope definition	• product line requirements • product line architecture documentation • product line architecture evaluation report • asset inventory • mining plan and process • mined assets • commercial off-the-shelf (COTS) criteria • COTS assets • core components • product line test strategy, test cases, test architecture, test scripts, and test plan • attached processes	• product requirements • product architecture documentation • product architecture evaluation report • product-specific components (mined, COTS, or built new) • product test strategy, test cases, test architecture, test plan
Process outputs	defined processes for • requirements engineering • project management • software configuration management • development • testing • risk management • architecture conformance	• configuration management process for product lines • tool support list • development tool set • production tool set • measurement plan • core asset metrics • core asset work plans • production plan	• product metrics • product work plans

Table 2: Outputs in the Adoption Phases (cont'd.)

	Establish Context Phase	Establish Production Capability Phase	Operate Product Line Phase
Organization outputs	• adoption plan • funding model • organization chart • product line concept of operations (CONOPS) • marketing plan • product proposals • acquisition strategy • organization risk management plan or process • training plan • product line training	• progress reports • risks and mitigation strategies	• organizational metrics • cost/pricing model • product release strategy • trouble reports • customer feedback • upgraded plans • improvement suggestions • risks and mitigation strategies

For example, this view shows that, from a product perspective in the Establish Context Phase, an organization needs to get a handle on the characteristics of the set of products the core assets should address before any real emphasis is placed on defining the product line architecture. Other outputs may be warranted in particular product line adoption efforts, and some of those listed above may not apply in a given situation. Nonetheless, Table 1 can serve as a handy checklist for representative output from each phase.

3.2.6 Roles View

None of the views thus far depict the type of people who need to be involved in the product line adoption effort. Making task assignments requires such information. The Roles view, presented in Table 2 below, lists the typical roles associated with each quadrant of the Phases and Focus Areas view and identifies the operative roles in a product line adoption effort, organized to indicate staffing needs.

Table 3: Roles Involved in the Adoption Phases

	Establish Context	**Establish the Production Capability**	**Operate Product Line**
Product-related roles	• marketer • market analyst • domain expert • product manager • senior manager • technology scout • architect	core asset developer: • requirements engineer • architect • architecture evaluator • component developer • tester • software integrator	product developer: • requirements engineer • architect • architecture evaluator • component developer • tester • software integrator
Process-related roles	• technical manager • process owner • process group member	• technical manager • process owner • process group member • technical support • tool specialist • measurement specialist	• technical manager • process owner • process group member • technical support • tool specialist • measurement specialist
Organization-related roles	• product line manager • software manager • business unit or organization manager • product manager • acquisition expert • financial manager • human resource manager • training planner • training developer • trainer	• product line manager • software manager • business unit or organization manager • financial manager • training developer • trainer	• product line manager • product manager • business unit or organization manager • customer field representative • salesperson

This view clearly shows that a mixture of technical and business talent is required to adopt a product line approach. For example, marketers, managers, domain experts, technologists, and architects are all involved in establishing the product context, which involves defining the scope and its associated business case. Note that, though the same roles may appear in multiple phases, the tasks those roles perform will vary with the phase. For example, the training developer in the Establish Context Phase will be developing training for the management and technical personnel to understand and apply product line concepts and practices. The training developer in the Establish Production Capability Phase will be teaching developers—and potentially marketers or customers—to understand and use the core assets for the particular product line being fielded. As in the Outputs view, the actual roles performed in any specific product line adoption effort may differ slightly, but Table 2 is still a handy reference.

4 Using the Adoption Factory Pattern

The views of the Adoption Factory pattern presented in Section 3.2 make it easier to use the pattern as a generic product line adoption roadmap. The phases and focus areas become a natural overall organizing mechanism. The practice areas provide the more detailed information. The roles help identify who needs to be involved and the talent that may need to be acquired. The outputs provided a checklist of expected deliverables from each phase. Using these views, an organization can plan to master the necessary practice areas in a continuous way that begins at the phase where each practice areas first appears. For example, scoping (which is part of the What to Build pattern) is central to the Establish Context Phase and must begin early in the product line effort, though it continues throughout the rest of the product line adoption process (and life cycle). However, the "Component Development" practice area in the Product Parts pattern—though it's certainly important to the product line effort—can begin later in the adoption process. The organization should address other practice areas in a similar fashion, always bearing in mind that there is no cement wall between the phases. Rather, because of the inherent iteration in the product line process, there will always be some backwash to earlier phases.

The Adoption Factory pattern is a generic adoption roadmap. Like any generic artifact, it is missing details—those necessary for any specific organization to use the pattern as a path to product line adoption. An organization should, with prudent organizational insight, instantiate and customize the roadmap to meet its needs. An organization, knowing its own strengths, challenges, and timeline for adoption, should also look across the phase horizon and, where it makes sense, begin to prepare early for those activities presenting the greatest challenges. For example, if an organization will rely heavily on legacy assets, it should begin the inventorying part of the "Mining Existing Assets" practice area during the Establish Context Phase in conjunction with scoping. If an organization (as depicted in Scenario 2 in the pattern definition) dove into the Establish Production Capability Phase by starting with architecture definition, it will want to plan to backfill sufficiently with practice areas of the Establish Context Phase to ensure that the architecture is targeting a set of products that makes business sense.

The adoption strategy can be proactive (core assets are built before products in the product line), reactive (one or more products are built before the core asset base is established), or some combination of the two [Krueger 02]. The Adoption Factory pattern applies regardless of the approach chosen. The mapping is intuitive for the proactive approach. In a reactive approach, the Establish Context Phase happens in a micro sense before the first product or products are built as single systems and then in a more deliberate way once the decision is made to extract or reengineer a core asset base from this product or products. The Establish Production Capability Phase involves the extracting or reengineering of the production plan (covered under the Product Parts pattern or its Plowed Field variant pattern) and the development of the plan (covered under the Assembly Line pattern). The case study of Salion, Inc.—a company that chose a reactive adoption strategy—illustrates this progression [Clements & Northrop 02b].

4.1 Embedding the Adoption Factory Pattern in a Technology Change Model

Software product line adoption is a special case of a technology change project in that it helps organizations adopt a new technology or new way of doing business. All technology change involves assessing the current state, identifying the desired state, and bridging the gulf between the two. Technology change requires vision, skills, incentives, resources, and a plan. Technology change experts recommend a technology change project that is charged with the transformation. A technology change project for product lines may involve

- changing the way people think about system building
- instituting new practices and procedures
- designing new organizational interfaces (both internal and external)
- reorganizing the staff

In short, product line adoption involves putting all 29 practices described in the Framework into place in an appropriate manner befitting particular organizational situations.

Organizations often use technology change models to guide their technology change efforts. One such model is the IDEAL[SM] model [McFeeley 96]. As illustrated in Figure 8, the IDEAL Model has five stages:

1. Initiating
2. Diagnosing
3. Establishing
4. Acting
5. Learning

[SM] IDEAL is a service mark of Carnegie Mellon University.

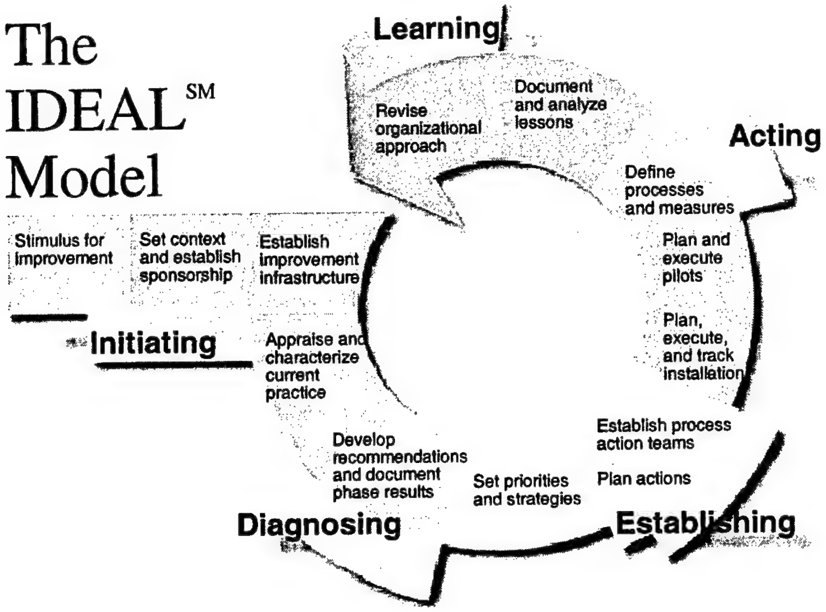

The IDEAL SM Model
Learning
Revise organizational approach
Document and analyze lessons
Acting
Define processes and measures
Plan and execute pilots
Plan, execute, and track installation
Stimulus for improvement
Set context and establish sponsorship
Establish improvement infrastructure
Initiating
Appraise and characterize current practice
Develop recommendations and document phase results
Set priorities and strategies
Establish process action teams
Plan actions
Diagnosing
Establishing

Figure 8: IDEAL Model

The Adoption Factory pattern can be overlaid on the more general IDEAL model as follows:

Initiating: Use the Adoption Factory pattern as an easily understood adoption vocabulary that can be shared across an organization and that marks organizational progress. Use the completion of phases or focus areas as product line adoption goals. Use the associated roles to guide staffing and management.

Diagnosing: Use the Adoption Factory pattern to gauge which phase of the adoption process an organization is in and to benchmark its activities by measuring them against the practice areas in that phase.[2]

Establishing: Use the incremental nature of the Adoption Factory pattern to structure a Product Line Adoption Plan. Use the subpatterns and their associated practice areas as the basis of subservient action plans.

Acting: Follow the plans that are based on the Adoption Factory pattern. Apply the practice areas in the organization focus area to steer and manage the activities.

Learning: Collect data and lessons learned in each phase of the Adoption Factory pattern as specified by the "Data Collection, Metrics, and Tracking" practice area. Analyze results against established goals. Iterate through the pattern phases and focus on different practice

[2] We use the Adoption Factory pattern in the SEI Product Line Quick Look (PLQL). (For more information on the PLQL, go to http://www.sei.cmu.edu/plp/plql.html.) We also use it in the PLTP: during analysis and when framing the PLTP's results. (For more information on the PLTP, go to http://www.sei.cmu.edu/plp/pltp.html.)

areas. In addition, modify artifacts, processes, and organizational structures to reflect lessons learned and to take advantage of potential optimizations.

4.2 Connecting the Adoption Factory Pattern to Plans Supporting Product Line Adoption

In any organization, there may be a hierarchical set of goals, strategies, and plans. Organizations usually decide to adopt a product line approach as a strategy to achieve specific business goals, so product line adoption is likely an articulated strategy in a business plan. Adopting a software product line then becomes the goal of a product line adoption plan, which describes how the necessary product line practices and activities are to be rolled out across the organization. The Product Line Adoption Plan begets other plans (often called action plans), which, in turn, have goals, strategies, and so forth. Figure 9 depicts such a hierarchy of plans.

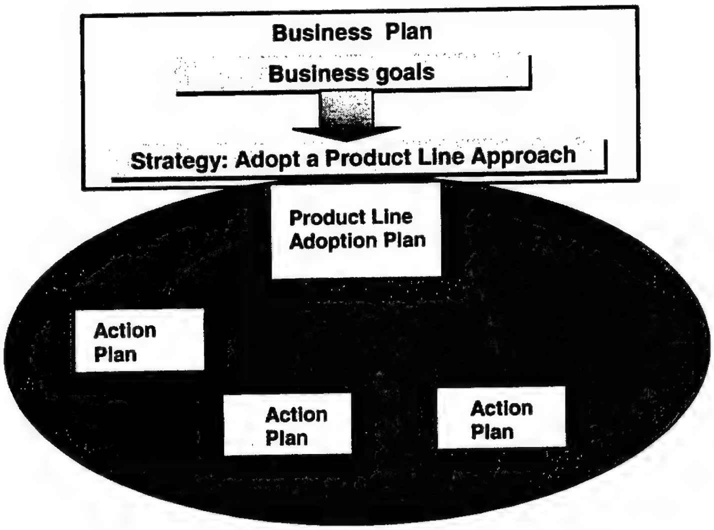

Business Plan
Business goals
Strategy: Adopt a Product Line Approach
Product Line Adoption Plan
Action Plan
Action Plan
Action Plan

Figure 9: Hierarchy of Plans

In Section 3, we discussed how the Adoption Factory pattern supports this whole gamut of plans. Table 3 summarizes the characteristics of the plans in the hierarchy and how the Adoption Factory pattern relates to them.

Table 4: Adoption Factory Connection to Organizational Plans

Type of Plan	Plan Characteristics	Connection to Adoption Factory
Business Plan	lays out overall company strategies to achieve business goals might specify adopting a software product line for a particular vertical segment of business	It's a prerequisite for using the Adoption Factory pattern. Its goals will serve as input to the product line business case.
Product Line Adoption Plan	describes how product line practices will be rolled out across the organization	The Adoption Factory pattern is used as an overall plan structure. Phases and focus areas become natural milestones. The Adoption Factory pattern is customized to fit organization-specific statuses, strengths, needs, and challenges.
Product Line Action Plan	addresses a specific portion of a product line adoption plan	It maps to a particular phase, focus area, subpattern, or practice area in the Adoption Factory pattern. Practice Areas, Roles, and Outputs views provide details for it.

5 Conclusion

Adopting a product line approach for fielding multiple similar software-intensive systems has proven to yield significant cost, quality, and competitive advantages for many organizations. However, adoption costs and barriers are often difficult to address, especially without explicit guidance. A generic, phased product line adoption roadmap has appeal. We have introduced the Adoption Factory pattern (a variant of the Factory pattern) and described how it can be used as a product line adoption roadmap. The Phases and Focus Areas, Practice Areas, Outputs, and Roles views provide the milestones, dependencies, and underlying detail that a technology adoption roadmap should have. We have described how the Adoption Factory pattern supports a proactive, reactive, or incremental approach to product lines. We have also detailed how the pattern would mesh with a general technology change model and underscored the pattern's ties to the various plans that an organization would create in adopting product lines.

At the SEI, we now use the Adoption Factory pattern routinely in our product line diagnostic instruments—the PLTP and its lighter weight counterpart, the Product Line Quick Look^{SM} (PLQL^{SM}). We use the Adoption Factory pattern to both analyze the collected organizational data and frame the diagnostic results. We are also using the pattern in our product line planning workshops, where we exploit its views to provide planning guidance and checklists for plan details. The organizations we have worked with have found the pattern to be helpful in understanding product line status and in planning and implementing their product line adoption efforts. They report that the pattern indeed provides a very useful vocabulary and abstraction that keeps even geographically distributed organizational counterparts on the same wavelength. Following a PLTP and a planning workshop that use the Adoption Factory pattern, one organization wrote about its successful results [Steger et al. 04]. We are most encouraged by our successful experiences and the positive feedback to date. We are now teaching the pattern and how to use it in our Adopting Software Product Lines course.

In conclusion, one cautionary note is important. Before embarking on a product line adoption course and using the Adoption Factory pattern to plot it, an organization must perform due diligence in examining the overall fit of a software product line approach. The following questions serve as good entry criteria for the pattern:

- Is there a multisystem business case?
- Does the organization have articulated goals it is trying to achieve with a software product line approach?

[SM] Product Line Quick Look and PLQL are service marks of Carnegie Mellon University.

- Do the benefits of successful product lines match the goals of the organization?
- Is there sufficient support within the organization to launch a software product line effort?

Next, we plan to do the following:

- develop some sample adoption plans and artifacts that illustrate more concretely the usefulness of the Adoption Factory pattern
- describe explicitly how we use the pattern in the PLTP
- explore the pattern's variants (e.g., for use in a Web services organization or a U.S. Department of Defense [DoD] acquisition organization)
- publish case studies that describe how specific organizations have applied the pattern

We welcome hearing about your experiences and receiving your feedback.

Appendix A Current Set of SEI Product Line Practice Patterns

The current SEI patterns and their variants are listed in Table 5.

Table 5: SEI Product Line Practice Patterns and Their Variants

Pattern	Variant
Assembly Line	none
Cold Start	Warm Start
Curriculum	none
Each Asset	Each Asset Apprentice Evolve Each Asset
Essentials Coverage	none
Factory	Adoption Factory
In Motion	none
Monitor	none
Process	Process Improvement
Product Builder	Product Gen
Product Parts	Green Field Barren Field Plowed Field
What to Build	Analysis Forced March

Appendix B Dynamic Structure of Selected Product Line Practice Subpatterns in the Factory Pattern

Figure 10 - Figure 18 below show the dynamic structure of the subpatterns in the Factory pattern and its variant, the Adoption Factory pattern.

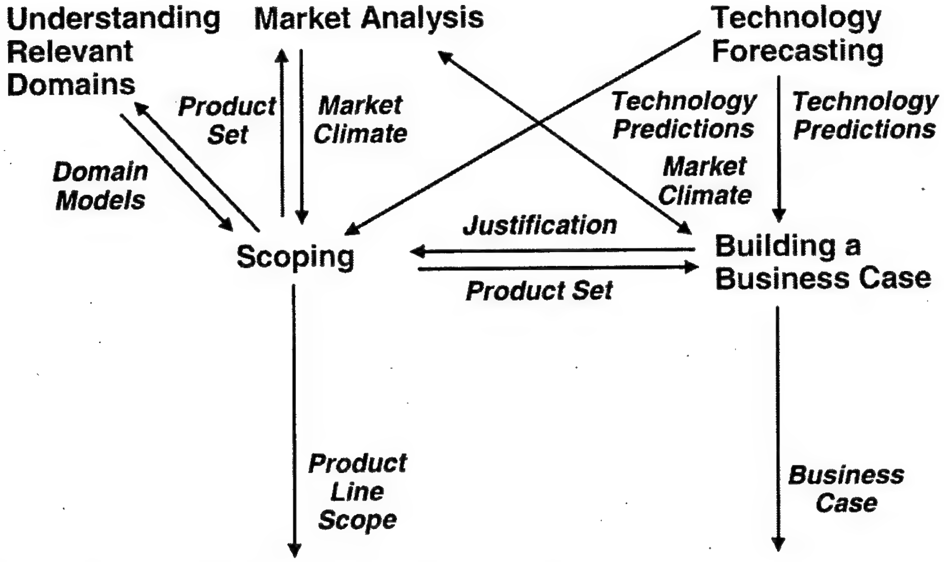


Figure 10: What to Build Pattern: Dynamic Structure

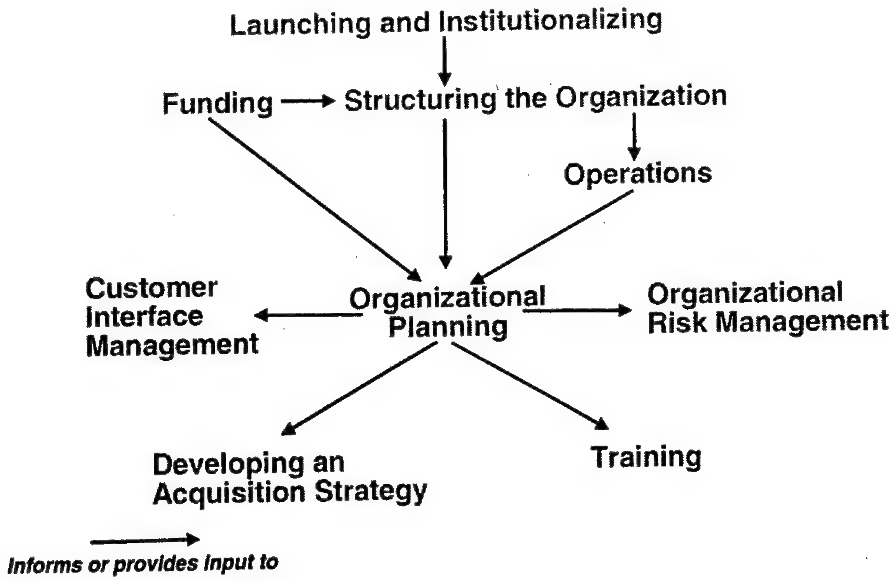

Launching and Institutionalizing
Funding
Structuring the Organization
Operations
Customer Interface Management
Organizational Planning
Organizational Risk Management
Developing an Acquisition Strategy
Training
Informs or provides input to

Figure 11: Cold Start Pattern: Dynamic Structure

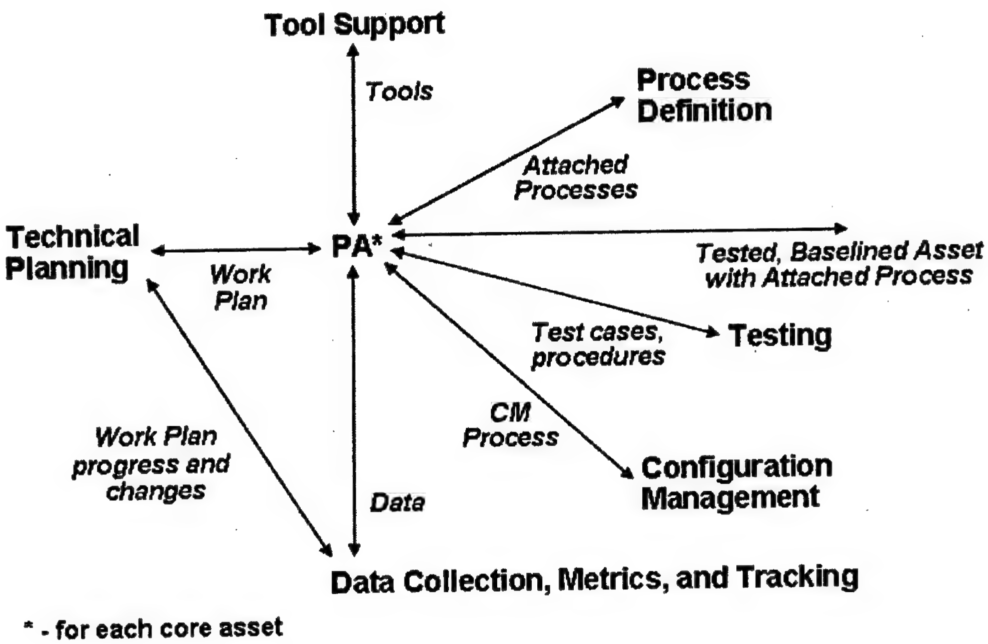


Figure 12: Each Asset Pattern: Dynamic Structure

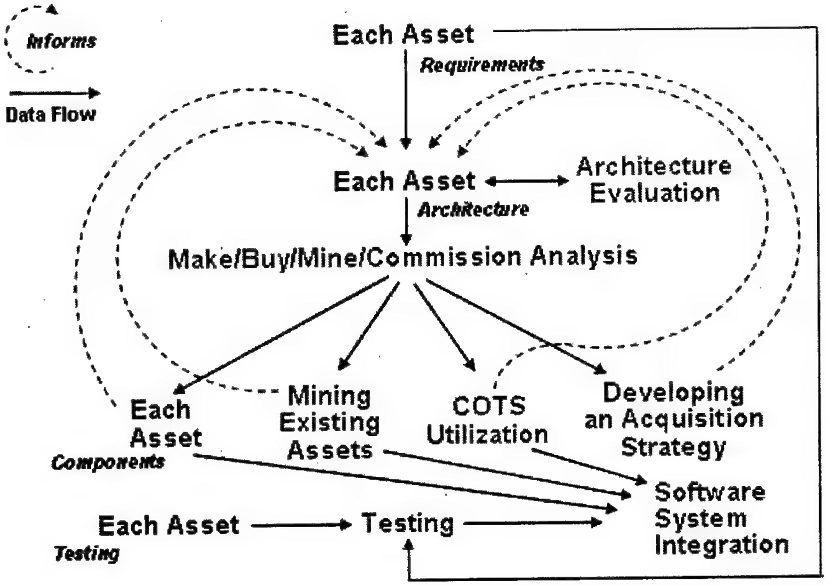


Figure 13: Product Parts Pattern: Dynamic Structure

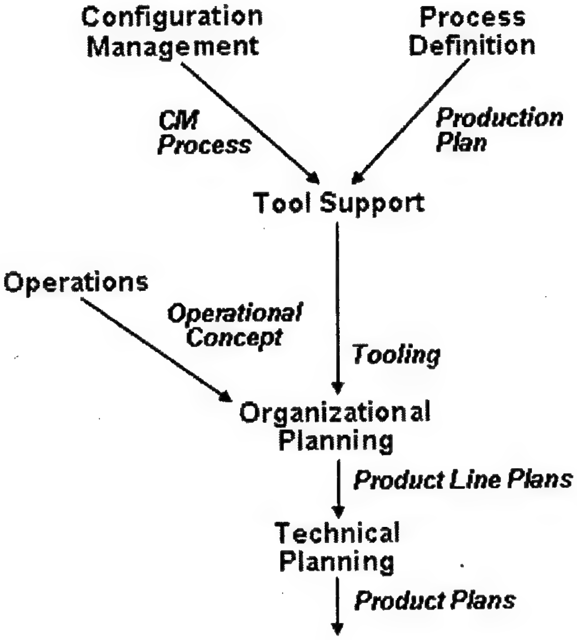


Figure 14: Assembly Line Pattern: Dynamic Structure

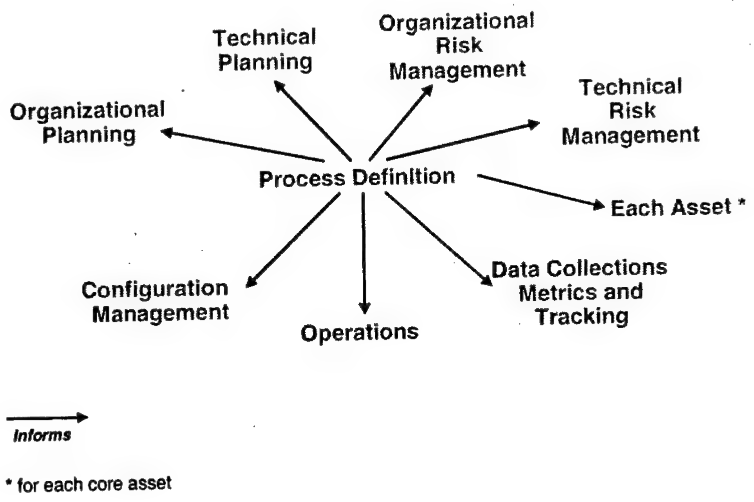


Figure 15: Process Pattern: Dynamic Structure

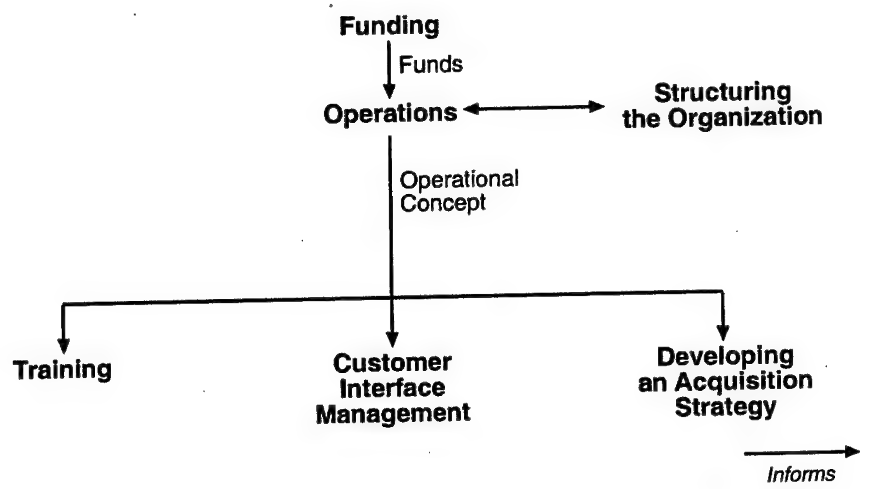

Funding
Funds
Operations
Structuring the Organization
Operational Concept
Training
Customer Interface Management
Developing an Acquisition Strategy
Informs

Figure 16: In Motion Pattern: Dynamic Structure

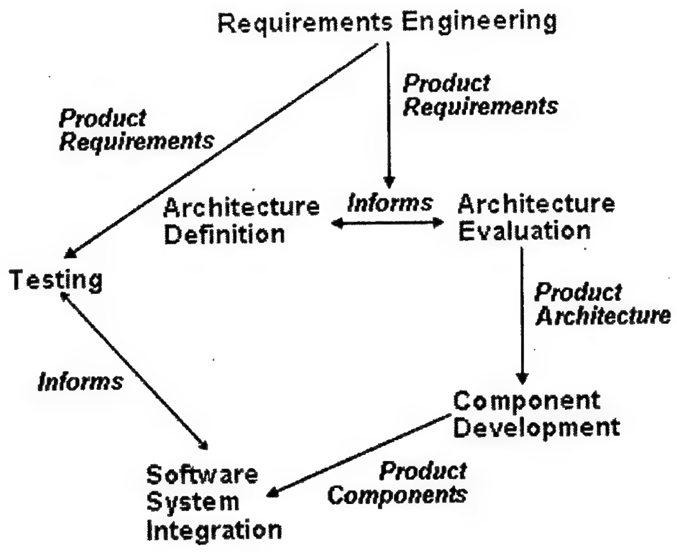


Figure 17: Product Builder Pattern: Dynamic Structure

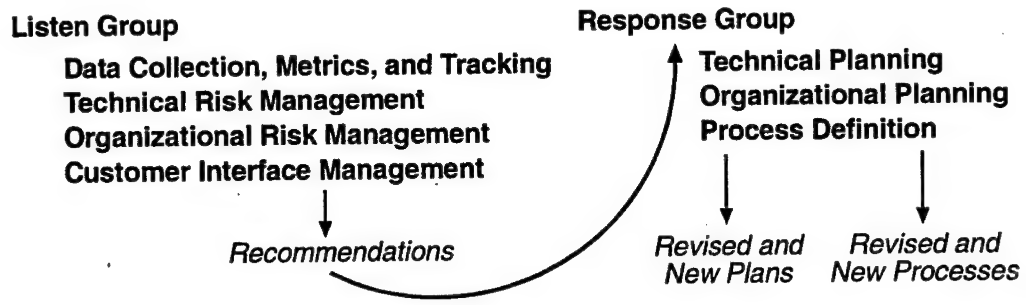

Listen Group
Data Collection, Metrics, and Tracking
Technical Risk Management
Organizational Risk Management
Customer Interface Management
Recommendations
Response Group
Technical Planning
Organizational Planning
Process Definition
Revised and New Plans
Revised and New Processes

Figure 18: Monitor Pattern: Dynamic Structure

References

[Böckle et al. 02] Böckle, G.; Munoz, J.; Knauber, P.; Krueger, C. W.; Sampaio do Prado Leite, J. C.; van der Linden, F.; Northrop, L.; Stark, M.; & Weiss, D. M. "Adopting and Institutionalizing a Product Line Culture," 49-59. *Software Product Lines: Proceedings of the Second International Software Product Lines Conference*. San Diego, CA, August 19-22, 2002. New York, NY: Springer-Verlag, 2002.

[Bosch 02] Bosch, J. "Maturity and Evolution in Software Product Lines: Approaches, Artefacts and Organization," 257-271. *Software Product Lines: Proceedings of the Second International Software Product Lines Conference*. San Diego, CA, August 19-22, 2002. New York, NY: Springer-Verlag, 2002.

[Bühne et al. 03] Bühne, S.; Chastek, G.; Kakola, T.; Knauber, P.; Northrop, L.; & Thiel, S. "Exploring the Context of Product Line Adoption." *Proceedings of the Product Family Engineering WorkshopPFE-5.* Sienna, Italy, November 4-6, 2003. Berlin, Germany: Springer-Verlag, 2004.

[Bruckhauas 96] Bruckhauas, T.; Madhavji, N. H.; Janssen, I.; & Henshaw, J. "The Impact of Tools on Software Productivity." *IEEE Software 13,* 5 (September 1996): 29–38.

[Clements & Northrop 02a] Clements, P. & Northrop, L. *Software Product Lines: Practices and Patterns*. Boston, MA: Addison-Wesley, 2002.

[Clements & Northrop 02b] Clements, P. & Northrop, L. *Salion, Inc.: A Software Product Line Case Study* (CMU/SEI-2002-TR-038, ADA412311). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu/publications/documents /02.reports/02tr038.html

[Clements & Northrop 04] *A Framework for Software Product Line Practice, Version 4.2.* http://www.sei.cmu.edu/plp/framework.html (August 2004).

[Clements et al. 04] Böckle, G; Clements, P.; McGregor, J. D.; Muthig, D.; & Schmid, K. "Calculating ROI for Software Product Lines." *IEEE Software 21*, 3 (June 2004): 23-31.

[Gamma 95] Gamma, E.; Helms, R.; Johnson, R.; & Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software.* Reading, MA: Addison-Wesley, 1995.

[Krueger 02] Krueger, C. W. "Easing the Transition to Software Mass Customization," 282-293. *Proceedings of the 4th International Workshop on Software Product Family Engineering.* Bilbao, Spain, October 3-5, 2001. Berlin, Germany: Springer-Verlag, 2002.

[McFeeley 96] McFeeley, R. *IDEAL: A User's Guide for Software Process Improvement* (CMU/SEI-96-HB-001, ADA305472). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996. http://www.sei.cmu.edu/publications/documents /96.reports/96.hb.001,html

[Muthig 02] Muthig, D. *A Light-Weight Approach Facilitating an Evolutionary Transition Towards Software Product Lines.* Stuttgart, Germany: Fraunhofer IRB Verlag, 2002 (ISBN 3816762085).

[Northrop 02] Northrop, L. "SEI's Software Product Line Tenets." *IEEE Software 19*, 4 (July/August 2002): 32-40.

[Schmidt & Verlage 02] Schmidt, K. & Verlage, M. "The Economic Impact of Product Line Adoption and Evolution." *IEEE Software 19*, 4 (July/August 2002): 50-57.

[Steger et al. 04] Steger, M.; Tischer, C.; Boss, B.; Müller, A.; Pertler, O.; Stolz, W.; & Ferber, S. "Introducing PLA at Bosch Gasoline Systems: Experiences and Practices," 34-50. *Software Product Lines: Proceedings of the Third International Software Product Lines Conference.* Boston, MA, August 30-September 2, 2004. Berlin, Germany: Springer-Verlag, 2004.

[van der Linden et al. 04] van der Linden, F.; Bosch, J.; Kamsties, E.; Känsälä, K.; & Obbink, H. "Software Product Family Evaluation," 110-129. *Software Product Lines: Proceedings of the Third International Software Product Lines Conference.* Boston, MA, August 30-September 2, 2004. Berlin, Germany: Springer-Verlag, 2004.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY
(Leave Blank)

2. REPORT DATE
September 2004

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Software Product Line Adoption Roadmap

5. FUNDING NUMBERS
F19628-00-C-0003

6. AUTHOR(S)
Linda M. Northrop

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-2004-TR-022

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ESC-TR-2004-022

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited, DTIC, NTIS

12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

The tremendous benefits of taking a software product line approach are well documented. Organizations have achieved significant reductions in cost and time to market and, at the same time, increased the quality of families of their software systems. However, to date, there are considerable barriers to organizational adoption of product line practices and to widespread product line practice. Phased adoption is attractive as a risk reduction and fiscally viable proposition. This report introduces a variant of the Factory Pattern called the Adoption Factory pattern that provides a generic roadmap to guide a manageable, phased product line adoption strategy. In addition, this report examines the Adoption Factory pattern from multiple useful views and describes how it can be used. This report concludes with a summary of the Carnegie Mellon® Software Engineering Institute's experiences with the pattern to date and its future plans with regard to the pattern.

14. SUBJECT TERMS
software product line, product line adoption, adoption roadmap, product line practice patterns

15. NUMBER OF PAGES
54

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL